IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Dale John Shidla, et al	Examiner: Dave, Jyoti D.
Serial No. 10/658,983	Art Unit: 4182
Filing Date: September 10, 2003	Attorney Docket No.: 200310483-1

Title: Compiler-Scheduled CPU Functional Testing

Honorable Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF FILED UNDER 37 C.F.R. § 41.37

Sir:

This appeal brief is being filed with a notice of appeal in response to a final office action mailed on August 12, 2008 for the above-referenced patent application.

The Commissioner is hereby authorized to charge requisite fees due for this submission to Deposit Account No. 08-2025 (Hewlett Packard).

I. REAL PARTY IN INTEREST

The real party in interest is the Hewlett-Packard Development Company, L.P., a Texas Limited Partnership having its principal place of business in Houston, Texas. The Hewlett-Packard Development Company, L.P., is the assignee of the present application.

II. RELATED APPEALS AND INTERFERENCES

On information and belief, there are no appeals, interferences, or judicial proceedings known to the appellant, the appellant's legal representative, or assignee which may be related to, directly affect or be directly affected by or have a bearing on the Board of Patent Appeals and Interferences (the "Board") decision in the pending appeal.

III. STATUS OF CLAIMS

A. Total Claims: 1-18

B. Current Status of Claims:

1. Claims canceled: none
2. Claims withdrawn: none
3. Claims pending: 1-18
4. Claims allowed: none
5. Claims rejected: 1-18
6. Claims objected to: none

C. Claims on Appeal: 1-18

As indicated above, claims 1-18 are pending in this application, stand finally rejected, and are being appealed. These claims are rejected in the final office action mailed August 12, 2008 ("the last office action").

IV. STATUS OF AMENDMENTS

No amendment has been filed after the final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The claimed subject matter relates to computers and computer software. More particularly, the claimed subject matter relates to software compilers for computers.

Independent claim 1 recites a method of compiling a program (see FIGS. 3a and 3b, and page 4, line 18 through page 5, line 18) to be executed on a target microprocessor with multiple functional units of a same type (page 2, lines 3-5; blocks 78A and 78B in FIG. 5 and page 7, lines 29-31; and blocks 92A and 92B in FIG. 6 and page 8, lines 23-29). The method comprises opportunistically scheduling a redundant operation on one of the functional units that would otherwise be idle during a cycle (page 2, lines 5-7; block 64 in FIG. 4 and page 6, lines 3-5).

Independent claim 5 recites a method of compiling a program (see FIGS. 3a and 3b, and page 4, line 18 through page 5, line 18) to be executed on a target microprocessor (page 2, lines 3-5). Identification is made of a cycle during which an operation is available for a first functional unit and no operation is available for a second functional unit (page 2, lines 8-11; block 62 in FIG. 4, and page 6, lines 1-3). The first and second functional units comprise functional units of a same type (page 2, lines 3-5; blocks 78A and 78B in FIG. 5 and page 7, lines 29-31; and blocks 92A and 92B in FIG. 6 and page 8, lines 23-29). The operation is scheduled for execution by both the first and second functional units during the cycle (page 2, lines 11-12; block 64 in FIG. 4 and page 6, lines 3-5). A comparison of results obtained by the first and second functional units is scheduled to be during a subsequent cycle (page 2, lines 12-14; block 66 in FIG. 4 and page 6, lines 5-9).

Independent claim 16 recites a program compiler (see FIGS. 3a and 3b, and page 4, line 18 through page 5, line 18) for a target microprocessor with multiple equivalent functional units (page 2, lines 15-16). The compiler comprises a code generator (block 42 in FIG. 3a) including a scheduler (block 52 in FIG. 3b) that opportunistically schedules a redundant operation on one of the functional units that would otherwise be idle during a cycle (page 2, lines 17-19; block 64 in FIG. 4 and page 6, lines 3-5).

Independent claim 18 recites a computer-readable program product for execution on a target microprocessor with multiple functional units of a same type (page 2, lines 20-22; blocks 78A and 78B in FIG. 5 and page 7, lines 29-31; and blocks 92A and 92B in

FIG. 6 and page 8, lines 23-29). The program product comprises executable code that includes a redundant operation scheduled for one of the functional units that would otherwise be idle during a cycle (page 2, lines 22-24; block 64 in FIG. 4 and page 6, lines 3-5) and also includes a subsequently scheduled comparison of results from the redundant operation for fault checking purposes (page 2, lines 24-26; block 66 in FIG. 4 and page 6, lines 5-9).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The following grounds of rejection are to be reviewed on appeal:

1. The rejection of claims 1-3 and 18 under 35 U.S.C. §103 as being unpatentable over Metzger (US 7,269,827) in view of Quach (US 6,640,313);

2. The rejection of claim 4 under 35 U.S.C. §103 as being unpatentable over Metzger (US 7,269,827) in view of Quach (US 6,640,313) and further in view of Fruehling et al. (US 6,625,688);

3. The rejection of claims 5-10 and 14-15 under 35 U.S.C. §103 as being unpatentable over Metzger (US 7,269,827) in view of Quach (US 6,640,313) and further in view of Raina (US 6,134,675);

4. The rejection of claims 11, 13 and 16-17 under 35 U.S.C. §103 as being unpatentable over Metzger (US 7,269,827) in view of Quach (US 6,640,313) in view of Raina (US 6,134,675) and further in view of Chan et al. (US 5,557,761); and

5. The rejection of claim 12 under 35 U.S.C. §103 as being unpatentable over Metzger (US 7,269,827) in view of Quach (US 6,640,313) in view of Raina (US 6,134,675) in view of Chan et al. (US 5,557,761) and further in view of Fruehling et al. (US 6,625,688).

VII. ARGUMENT

Applicants respectfully traverse the aforementioned rejections of claims 1-18 in the latest office action for the following reasons.

A. Claims 1-4

THE CITED ART DOES NOT DISCLOSE OR SUGGEST "OPPORTUNISTICALLY SCHEDULING A REDUNDANT OPERATION ON ONE OF THE FUCNTIONAL UNITS THAT WOULD OTHERWISE BE IDLE DURING A CYCLE"

Claim 1 recites as follows.

> 1. A method of compiling a program to be executed on a target microprocessor with multiple functional units of a same type, the method comprising **opportunistically scheduling a redundant operation on one of the functional units that would otherwise be idle during a cycle.**
>
> (Emphasis added.)

As shown above, claim 1 recites "A method of compiling a program to be executed on a target microprocessor with **multiple functional units of a same type**, the method comprising **opportunistically scheduling a redundant operation on one of the functional units that would otherwise be idle during a cycle.**" Applicants respectfully submit that this claim limitation is not disclosed or suggested by Metzger in view of Quach.

In regard to Metzger, as admitted in the first office action (dated February 7, 2008), Metzger does not disclose compiling for a target microprocessor **with multiple functional units of a same type**. (First office action, page 4, line 4.) Applicants agree with this observed deficiency in Metzger. Given that Metzger does not disclose compiling for a target microprocessor **with multiple functional units of a same type**, applicants respectfully submit that Metzger cannot be reasonably interpreted to disclose the **opportunistic scheduling of a redundant operation on one of the multiple functional units of a same type**.

Regardung Quach, the Examiner makes two incorrect assertions regarding the disclosure of Quach.

First, the Examiner asserts that column 6, lines 1-2 of Quach "clearly discloses a method that **tests functional units** which comprise of logic units of the same type" (first office action, page 4, lines 4-6, emphasis added). Applicants respectfully traverse this assertion. Column 6, lines 1-2 of Quach merely states as follows. "In HR mode, issue unit 120 provides identical instructions to execution clusters 140(a) and 140(b)." However, the issuance of identical instructions in high reliability (HR) mode is not to "test functional units," rather Quach discloses that the issuance of identical instruction in HR mode is to **detect soft errors**. (Abstract, lines 6-9, "The issue unit provides the same instructions to the execution clusters in high reliability mode and results generated by the different execution clusters are compared to detect soft errors."; col. 3, lines 23-39, "The present invention provides a flexible approach to mitigating the effects of soft errors in a processor .... For these critical code types, the hardware cost of implementing redundant execution is more than balanced by the elimination of system crashes that might otherwise occur."). Thus, the Examiner's interpretation that Quach "clearly discloses a method that **tests functional units** which comprise of logic units of the same type" is incorrect.

Second, the Examiner asserts that "Quach clearly teaches that the above method is better suited for **detecting errors** in a processing environment, **without creating more work for the processor** (Quach column 3, lines 22-25)." (Emphasis added.) Applicants respectfully traverse this assertion. Column 3, lines 23-25 of Quach recite as follows. "The present invention provides a flexible approach to **mitigating the effects of soft errors** in a processor **without increasing the processor's chip area** significantly." (Emphasis added.) Hence, the interpretation of Quach by the Examiner again appears to be incorrect.

Thus, the Examiner's interpretations of Quach appear to be misleading and incorrect. Hence, applicants respectfully submit that the reasoning given by the Examiner for the obviousness rejection of claim 1 is flawed and that a proper *prima facie* case for rejecting claim 1 has not been made.

Claims 2-4 depend from claim 1 and so also overcome their rejections.

Applicants note that the citation to Fruehling et al. does not cure the above-discussed deficiencies in Metzger in view of Quach. Fruehling et al. is cited in relation to claim 4 as teaching "different levels or modes the PSA can work in." (Paragraph 12 on page 7 of the first office action.)

B. Claims 16-17

Similar to claim 1, claim 16 recites, "a code generator including **a scheduler that opportunistically schedules a redundant operation on one of the functional units that would otherwise be idle during a cycle**." Hence, for at least the same reasons discussed above in Section A in relation to claim 1, applicants respectfully submit that claim 16 also overcomes its rejection.

Applicants note that the citations to Raina and Chan et al. do not cure the above-discussed deficiencies in Metzger in view of Quach. Raina is cited in relation to claim 13 for disclosing a cross compiler (paragraph 36 on page 14 of the first office action), and Chan et al. is cited in relation to claim 16 for disclosing a code generator of a compiler (paragraph 39 on page 16 of the first office action).

Claim 17 depends from claim 16 and so also overcomes its rejection.

C. Claim 18

Similar to claim 1, claim 18 recites, "executable code that includes **a redundant operation scheduled for one of the functional units that would otherwise be idle during a cycle**...." Hence, for at least the same reasons discussed above in Section A in relation to claim 1, applicants respectfully submit that claim 18 also overcomes its rejection.

D. Claims 5-15

THE CITED ART DOES NOT DISCLOSE OR SUGGEST ALL THE ELEMENTS OF CLAIM 5

Claim 5 recites as follows.

> 5. A method of compiling a program to be executed on a target microprocessor, the method comprising:
> identifying a cycle during which an operation is available for a first functional unit and no operation is available for a second functional unit, wherein the first and second functional units comprise functional units of a same type;
> **scheduling the operation for execution by both the first and second functional units during the cycle**; and
> **scheduling a comparison of results obtained by the first and second functional units during a subsequent cycle.**
>
> (Emphasis added.)

As shown above, claim 1 recites both "scheduling the operation for execution by both the first and second functional units during the cycle" and "scheduling a comparison of results obtained by the first and second functional units during a subsequent cycle." Applicants respectfully submit that this claim limitation is not disclosed or suggested by Metzger in view of Quach and further in view of Raina.

In regard to Metzger, as admitted in the first office action, Metzger does not disclose "scheduling the operation for execution by both the first and second functional units during the cycle" and "scheduling a comparison of results obtained by the first and second functional units during a subsequent cycle." (First office action, paragraph 15 on page 9.) Applicants agree with these observed deficiencies of Metzger.

Regardung Quach, the Examiner makes an incorrect assertion regarding the disclosure of Quach. Specifically, the Examiner asserts in paragraph 16 of the first office action that column 5, lines 44-49 of Quach discloses "**scheduling the operation for execution by both the first and second functional units during the cycle**" (emphasis in

original). Applicants respectfully traverse this assertion. Column 5, lines 44-49 merely states that "multi-porting allows resources in execution clusters 140(a) and 140(b) to access the same register entry simultaneously in HR mode." There appears to be no disclosure in the citation in regards to scheduling the identified operation for execution by both units. Thus, the Examiner's interpretation that Quach discloses "**scheduling the operation for execution by both the first and second functional units during the cycle**" (emphasis in original) is incorrect.

Regarding Raina, the Examiner also makes an incorrect assertion regarding the disclosure of Raina. Specifically, the Examiner asserts in paragraph 17 of the first office action that column 2, lines 7-13 of Raina discloses "**scheduling a comparison of results obtained by the first and second functional units during a subsequent cycle**" (emphasis in original). Applicants respectfully traverse this assertion. Column 2, lines 4-16 states as follows.

> Referring to FIG. **1**, an embodiment of a system **10** that uses a testing device **12** is disclosed. The system **10** includes a plurality of microprocessor cores **14**, **16**, **18**, and **20** to be tested by the testing device **12**. The testing device **12** receives a core selection **22** and a plurality of input signals **24**, **26**, **28** and **30** from the plurality of processor cores **14**, **16**, **18** and **20**. The testing device **12** produces a test result output **32** that is to be compared with an expected logic value using an external tester **70** to determine a testing result **76**. While the particular exemplary embodiment of FIG. **1** illustrates four processor cores, it is to be understood that the present invention is applicable to testing a plurality of processor cores .

The testing device 12 is hardware circuitry shown in FIG. 1. There appears to be no disclosure in the citation in regards to scheduling a comparison of results. Thus, the Examiner's interpretation that Raina discloses "**scheduling a comparison of results obtained by the first and second functional units during a subsequent cycle**" (emphasis in original) is incorrect.

Thus, the Examiner's interpretations of Quach and Raina appear to be misleading and incorrect. Hence, applicants respectfully submit that the reasoning given by the

Examiner for the obviousness rejection of claim 5 is flawed and that a proper *prima facie* case for rejecting claim 5 has not been made.

Claims 6-15 depend from claim 5 and so also overcome their rejections.

Applicants note that the citations to Chan et al. and Fruehling et al. do not cure the above-discussed deficiencies in Metzger in view of Quach and further in view of Raina. Chan et al. is cited in relation to claim 11 as disclosing a code generator of a software compiler (paragraph 34 on page 14 of the first office action), but not in regards to scheduling the identified operation for execution by both units and scheduling a comparison of the results. Fruehling et al. is cited in relation to claim 4 as teaching "different levels or modes the PSA can work in" (paragraph 12 on page 7 of the first office action).

VIII. CONCLUSION

For the above-discussed reasons, applicants respectfully request that the rejections of claims 1-18 be overturned.

Respectfully submitted,
DALE JOHN SHIDLA, ET AL.

Dated: October 6, 2008

/James K. Okamoto, Reg. No. 40,110/
James K. Okamoto, Reg. No. 40,110
Okamoto & Benedicto LLP
P.O. Box 641330
San Jose, CA 95164
Tel.: (408) 436-2110
Fax.: (408) 436-2114

CERTIFICATE OF MAILING			
I hereby certify that this correspondence, including the enclosures identified herein, is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below. If the Express Mail Mailing Number is filled in below, then this correspondence is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service pursuant to 37 CFR 1.10.			
Signature:			
Typed or Printed Name:		Dated:	10/06/2008
Express Mail Mailing Number (*optional*):			

CLAIMS APPENDIX

CLAIMS INVOLVED IN THE APPEAL

1. A method of compiling a program to be executed on a target microprocessor with multiple functional units of a same type, the method comprising opportunistically scheduling a redundant operation on one of the functional units that would otherwise be idle during a cycle.

2. The method of claim 1, further comprising:
scheduling a comparison of results from the redundant operation.

3. The method of claim 2, further comprising:
causing a flag in the target microprocessor to be set when the comparison indicates an error.

4. The method of claim 2, further comprising:
setting a user-selectable level for an aggressiveness of said opportunistic scheduling.

5. A method of compiling a program to be executed on a target microprocessor, the method comprising:
identifying a cycle during which an operation is available for a first functional unit and no operation is available for a second functional unit, wherein the first and second functional units comprise functional units of a same type;
scheduling the operation for execution by both the first and second functional units during the cycle; and
scheduling a comparison of results obtained by the first and second functional units during a subsequent cycle.

6. The method of claim 5, wherein the first and second functional units comprise first and second floating point units of the target microprocessor.

7. The method of claim 5, wherein the first and second functional units comprise first and second arithmetic logic units of the target microprocessor.

8. The method of claim 5, wherein the results of the execution are stored in registers within the microprocessor, and the comparison of results compares contents of those registers.

9. The method of claim 5, wherein the target microprocessor includes at least three functional units of the same type.

10. The method of claim 9, further comprising:

identifying that during the cycle a second operation is available for a third functional unit of the same type and no operation is available for a fourth functional unit of the same type;

scheduling the second operation for execution by both the third and fourth functional units during the cycle; and

scheduling a comparison of results obtained by the third and fourth functional units during a subsequent cycle.

11. The method of claim 5, wherein the method is performed by a scheduler in a code generator of a program compiler.

12. The method of claim 11, wherein the program compiler comprises a native compiler for the target microprocessor.

13. The method of claim 11, wherein the program compiler comprises a cross compiler run on a different microprocessor.

14. The method of claim 5, further comprising:
causing a flag to be set when the comparison indicates an error.

15. The method of claim 14, further comprising:
if the error flag is set, then halting the execution and causing a notification to the user of the error flag.

16. A program compiler for a target microprocessor with multiple equivalent functional units, the compiler comprising a code generator including a scheduler that opportunistically schedules a redundant operation on one of the functional units that would otherwise be idle during a cycle.

17. The compiler of claim 16, wherein the scheduler subsequently schedules a comparison of results from the redundant operation.

18. A computer-readable program product for execution on a target microprocessor with multiple functional units of a same type, the program product comprising executable code that includes a redundant operation scheduled for one of the functional units that would otherwise be idle during a cycle and also includes a subsequently scheduled comparison of results from the redundant operation for fault checking purposes.

EVIDENCE APPENDIX

There are no documents or items submitted under this section.

RELATED PROCEEDINGS APPENDIX

There are no documents or items submitted under this section.